World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:16, No:3, 2022

Multi-Sensor Target Tracking Using Ensemble Learning

Bhekisipho Twala, Mantepu Masetshaba, Ramapulana Nkoana

***Abstract*—**Multiple classifier systems combine several individual classifiers to deliver a final classification decision. However, an increasingly controversial question is whether such systems can outperform the single best classifier, and if so, what form of multiple classifiers system yields the most significant benefit. Also, multi-target tracking detection using multiple sensors is an important research field in mobile techniques and military applications. In this paper, several multiple classifiers systems are evaluated in terms of their ability to predict a system's failure or success for multi-sensor target tracking tasks. The Bristol Eden project dataset is utilised for this task. Experimental and simulation results show that the human activity identification system can fulfil requirements of target tracking due to improved sensors classification performances with multiple classifier systems constructed using boosting achieving higher accuracy rates.

***Keywords*—**Single classifier, machine learning, ensemble learning, multi-sensor target tracking.

I. Introduction

TO a degree never known before, decision-makers have access to a vast amount of data. However, to use this information potential, the real-time data streams must not overwhelm the human beings involved. On the contrary, the data must be fused to high-quality information to provide decision support on various hierarchy levels. Being a challenging exploitation technology at the standard interface of sensors, command and control systems, and the human decision-makers, data and information fusion have enormous potential for innovative, intelligent, surveillance, reconnaissance systems in defence and civilian applications. One such application is the integration and fusion of multi-sensor in intelligent systems.

The topic of multisensory tracking has been of interest to researchers for more than 30 years. However, the development and successful fielding of multi-sensor tracking systems have lagged significantly behind the research activities despite recent work by [34]. Using a multi-sensor fusion approach with heterogeneous sensors, the information available for tracking depends on the sensors detecting the object [23].

A central concern of these applications is the need to increase the predictive accuracy of the tracking decision. An improvement inaccuracy or even a fraction of a percentage translates into significant future savings in time and costs. Thus, there has been an explosion of papers in the machine learning (ML) and statistical pattern recognition communities discussing how to combine models or model predictions in recent years.

Many works in both communities have shown that combining (ensemble) individual classifiers effectively improves predictive accuracy [2], [3], [9]. In this paper, the performance of several multiple classifier systems is evaluated in terms of their ability to predict a classifier system's failure or success in multisensory tracking.

An ensemble is generated by training multiple classifier systems for the same task. Their predictions are then combined to give an overall predictive accuracy rate. These ensembles can be developed in various ways with the resulting output combined for classification (for categorical response class attribute) and prediction (for numerical class attribute) tasks. The most popular training set resampling and variance reduction approach for ensemble learning includes changing the cases used for training through techniques such as bagging [2], [3], boosting [9], stacking [32], changing the features used in training [7], introducing randomness in the classifier itself [11].

Several multi-target tracking techniques have been implemented in the literature, including the issue of multiple classifier system methods for human activity identification by utilising diverse multimodal sensor data and classification algorithms [16], [19]-[21]. Specifically, these studies only developed protocols to integrate multiple accelerometer sensors attached at different body locations, limiting their robust activity recognition implementation. Other related works have fused information for activity recognition [22], [33], [36], while others have utilised smart wireless sensors in an intelligent system [23], [29] and biologically inspired methods in human and machine vision [30].

The paper's contribution is the proposal to use single classifier learning (SCL) systems for multi-sensor target tracking (MSTT) to help systems engineers execute their tasks. To find out if it would be worthwhile to overcome the limitations of SCL and their inability to handle more complex tracking situations, we then propose using multiple classifier learning (MCL) systems to deal with the MSTT problem. For the ensembles to increase the performance over that of the individual models, the unique models must be accurate individually, and they need to be sufficiently diverse. In other words, they need to be sufficiently different from each other in terms of which errors were made. For this reason, all possible combinations of the number of classifiers per ensemble are explored in this paper (i.e., from two

Bhekisipho Twala is with the Faculty of Engineering & the BE, Durban University of Technology, Durban, 4001 South Africa (e-mail: bhekisiphot@dut.ac.za).

Mantepu Masetshaba is with the Department of Decision Sciences, University of South Africa, Pretoria 0003, South Africa (e-mail: emasetmt@unisa.ac.za).

Ramapulana Nkoana is with the CSIR, Meiring Naude, Pretoria, South Africa (e-mail: rnkoana@csir.co.za).

classifiers per ensemble to five classifiers).

To this end, the rest of the paper is organised as follows. We place our work into past work in Section II and Section III, primarily single classifier systems and later multiple classifier systems. Section IV describes the experimental setup and learning algorithms. Then in Section V, we discuss the results presenting the corresponding conclusions.

II. Single Classifiers

Five base methods of classifier construction were considered for our simulation study: artificial neural network (ANN), algorithm quasi (AQ) rule induction system, decision tree (DT), *k*-nearest Neighbour (*k*-NN), Naïve Bayes classifier (NBC) and support vector machine (SVM). The five methods are briefly described below.

A. Artificial Neural Network

ANNs [25], [26] are usually non-parametric approaches (no assumptions about the data are made). They are represented by connections between a vast number of simple computing processors or elements (neurons). They have been used for a variety of classification and regression problems. The ANN is trained by supplying it with many numerical observations or the patterns to be trained (input data pattern) whose corresponding classifications (desired output) are known. During training, the final sum-of-squares error over the validation data for the network is calculated. Then, the selection of the optimum number of hidden nodes is made based on this error value. Once the network is trained, a new object is classified by sending its attribute values to the network's input nodes, applying the weights to those values, and computing the values of the output units or output unit activations. The assigned class is the one with the most significant output unit activation.

B. Algorithm-Quasi Optimal

A rule induction version of the Algorithm Quasi-optimal (AQ) was proposed initially by Michalski et al. [20] and further improved in later research work [20]. This robust ML methodology's implementation is based on a concept of a local (object-relative) decision reduct from rough set theory. In AQ, *A* is the set of all the characteristics $A1, A2, ..., An,$ whereby the seed is a concept member, a positive case. A selector is an expression relating to a variable that is a characteristic or a decision regarding the value of a variable, such as a contradiction of values. The main inspiration of the AQ algorithm is the generation of cover for every concept while calculating stars and choosing single complexes for the cover from those stars. According to [4], the AQ algorithm requires calculating conjuncts of incomplete stars. AQ has also been applied to solve several problems, including individuals within an evolutionary computation framework.

C. Decision Tree

The DT algorithm is a supervised learning algorithm that can be used for both classification and prediction problems [6], [24]. DT is similar to a tree, which starts with the root node and expands into branches thereby constructing a tree-like structure. DT classifiers are composed of internal nodes representing features of a dataset with branches representing decision rules and the leaf nodes representing the outcome. belongs to the family of supervised learning algorithms. DT classifiers have four primary objectives. According to [27], these are to 1) classify correctly as much of the training sample as possible; 2) generalise beyond the training sample so that unseen samples could be classified with as high accuracy as possible; 3) be easy to update as more training samples become available (i.e., be incremental); 4) and have as simple a structure as possible. Objective 1) is highly debatable and, to some extent, conflicts with objective 2). Also, not all tree classifiers are concerned with objective 3). DTs are non-parametric, and a valuable means of representing the logic embodied in software routines. A DT takes as input a case or example described by a set of attribute values and outputs a Boolean or multi-valued "decision". For this paper, we shall stick to the Boolean case.

D. *k*-Nearest Neighbour

Instance-based learning (also known as the *k*-NN algorithm) is one of the most state-of-the-art yet simple ML algorithms used for classification and prediction tasks [1]. The nearest neighbour (NN) algorithm works by assigning an unclassified sample point to classify the nearest of a set of previously classified points. First, with the entire training set and test set stored in the memory, the *k* points in the training data that are closest to the test value are identified and the distance between all the categories is calculated. The Euclidean, Manhattan or Hamming distance measures are normally used for this task. The Euclidean distance which is computed between the instance and each stored training instance whereby the new instance is assigned the class of the nearest neighbouring instance is the most commonly used method. These *k*-NNs are then computed, and the new instance is assigned the most frequent class among the *k* neighbours.

E. Support Vector Machine

The SVM is another supervised machine algorithm that can be used for classification or prediction tasks and even outlier detection [5], [8], [13]. The SVM algorithm creates the best line or decision boundary that segregates *n*-dimensional space into classes so that a new future data point (or vector) can be put in the correct category. The best decision boundary is referred to as a hyperplane. Thus, SVM finds a hyperplane in an *n*-dimensional space (where *n* is the number of features you have) with the value of each feature being the value of a particular coordinate that distinctly classifies the data points (vectors). Classification is performed by separating the two classes of data points from which many possible hyperplanes are chosen. The objective is to find a plane with the maximum margin (i.e., the maximum distance between data points of both classes). Maximising the margin distance provides some reinforcement so that future data points can be classified with more confidence.

III. Multiple Classifier Systems

Multiple classifier systems can be classified into one of three architectural types: 1) dynamic classifier selection (DCS); 2) multi-stage (MS); and 3) static parallel (SP). From the three, SP is probably the most commonly used architecture. Two or more classifiers are developed independently and in parallel [35]. The outputs from each classifier are then combined to deliver a final classification decision (where the decision is selected from a set of possible class labels). A large number of combination functions are available. These include majority voting, weighted majority voting, the product or sum of model outputs, the minimum rule, the maximum rule and Bayesian methods [10], [31].

For dynamic selection of classifiers, a single classifier is selected for each test sample with the goal of finding a subset of classifiers to classify the unknown instance. The highest level of competence is determined by computed means using several methods such as *k*-NN, clustering or (in some cases) multiple training datasets. Based on global performance measures, the dominance of one classifier does not necessarily imply entire dominance over all other classifiers. Weaker competitors will sometimes beat the overall best across some regions [15]. DCS problems are typically approached from a global and local accuracy perspective [17].

The second type of architecture is MS, where the classifiers are constructed iteratively. The parameter estimation process depends on the classifier's classification properties from previous stages at each iteration. Some MS approaches generate models that are applied in parallel using the same type of combination rules used for SP methods. For example, most boosts create weak classifiers combined to make stronger ones [28].

To set up an ensemble learning method (i.e., multiple classifier systems), we first need to select our base models (also known as weak learners) to be aggregated. Then, several primary meta-algorithms aim to combine vulnerable learners, including bagging, boosting, feature selection, randomisation, and stacking. For the paper, we only consider bagging, boosting and randomisation as our data partitioning techniques. While bagging and boosting is based on manipulating the training data gave a "base learning algorithm", for randomisation, the ensemble is created by randomising the internal decisions made by the "base" algorithm.

IV. Experimental Design

For the simulation study, five base methods of classifier construction were chosen. Each approach utilises a different form of parametric estimation/learning; between them, they generate various model forms: linear models, density estimation, trees and networks, and they are all practically applicable within multisensory tracking environments, with known examples of their application within the systems engineering industry.

First, single classifiers were constructed using each state-of-the-art classification method utilising MATLAB software [19]. These were used to provide benchmarks against which various multiple classifier systems were assessed. It was evident that the benefits of using ensembles could not be achieved by simply copying an individual model and combining the individual predictions. For this reason, all possible combinations of the number of classifiers per ensemble were explored (i.e., from two classifiers per ensemble to five classifiers). We define these ensembles as ENS5 (for all five classifiers in the ensemble), ENS4 (for four classifiers per ensemble), ENS3 (for three classifiers per ensemble) and ENS2 (for two classifiers per ensemble). The reader should note that within each ensemble, there are different combinations of single classifiers. For example, ENS4 has three different combinations; ENS3 has six different varieties, and ENS2 has 10 different combinations.

To measure the performance of classifiers, the training set/ test set methodology is employed. The Bristol Eden project dataset [18] is utilised for this task. Primarily, the data include scenarios of short-range surveillance type applications filmed under varying illumination conditions. The various scenes have people (dressed in civilian dress and camouflage, stationary, walking or running, or carrying multiple objects), vehicles, foliage and buildings/structures. Each dataset is split randomly into 80% training set and 20% testing or validating set for each run. The performance of each classifier is then assessed on the smoothed classification error rate (which has been shown to reduce bias and handles well the issue of a tie between two competing classes).

The fixed-effect model [12], [14] is used to test for statistical significance of the main effects (i.e., the five single classifiers; twenty multiple classifier systems, three multiple classifier architecture and three resampling procedures) versus their respective interactions. Each experiment is replicated five times making it a total of 5 x 20 x 3 x 3 x 5 = 4500 experiments.

V. Experimental Results & Findings

The results are summarised in Figs. 1-4 regarding smoothed error rates against the baseline classifiers (BASE) and their respective ensembles (ENS5, ENS4, ENS3, ENS2). DCSs that look to segment the population in several sub-regions are consistently poor performers, with all the experiments yielding inferior results to the single best classifier. However, the performance of most SP and MS combination strategies provides statistically significant improvements over the single best classifier.

From Fig. 1, it follows that DT is the best technique with a smoothed error rate of 29.6%. The second best method is *k*-NN, followed by ANN and SVM with smoothed error rates of 31.1%, 33.7%, and 34.1%. Finally, the worst performance is by AQ, with a smoothed error rate increase of 35.8%.

The results of the interaction effect (which was found to be statistically significant at the 5% level of significance) between MCL systems, architectures and resampling procedures are displayed in Figs. 2-5. It follows that all multiple classifier systems perform differently from each other, with significant smoothed error rate increases observed for ensembles with five or four classifiers compared to those with three or two classifiers per ensemble.

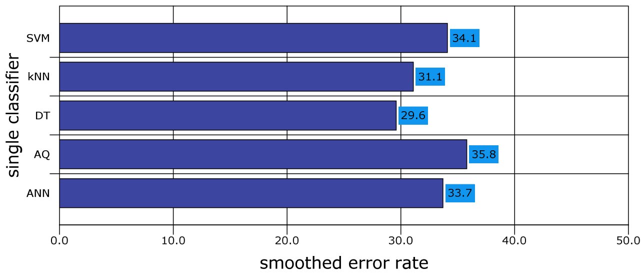

Fig. 1 Single classifiers

The results summarised in Fig. 2 show multiple classifier systems achieving higher accuracy rates when bagging is used as a sampling procedure followed by boosting and randomisation, respectively. Also, the experimental results of the three architectures used when constructing MCL systems show static-parallel exhibiting higher error rates than dynamic classifier systems and MS design. However, this is the only ensemble (ENS5) that outperformed individual classifiers for target tracking purposes, especially for static-parallel architecture.

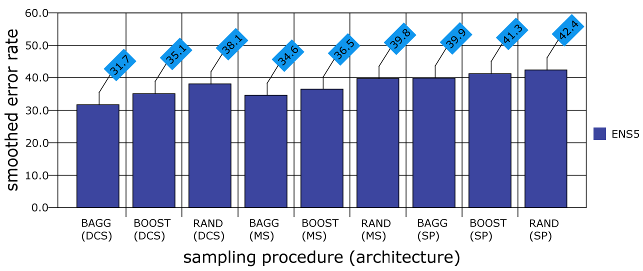

Fig. 2 Multiple classifier systems (5 members)

From Fig. 3, the effect of the sampling procedures on learning an ensemble with only four classifier members (i.e., ENS4) is prominent. The best overall performance is when bagging is used with a dynamic classifier system architecture, while the worst performance is when randomisation is used with SP architecture. In contrast, the best performance is observed when the ANN, AQ, DT, and SVM are the ensemble's four components. It also appears that the combination of AQ, DT and SVM makes the ensemble more accurate. Nonetheless, an ANN ensemble, AQ, DT, and *k*-NN exhibit the worst performance.

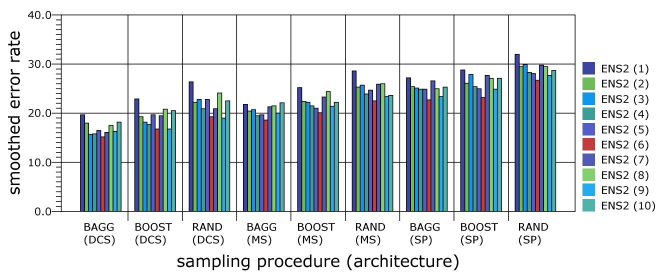

Fig. 3 Multiple classifier system (4 members)

For the DCS system, good performances by bagging and boosting are observed. At the same time, randomisation continues to struggle and achieves the worst performance, especially when AQ, DT and SVM are components of the ensemble and bagging is used as a sampling procedure (Fig. 4). The best performing ensemble is when the ANN, AQ and SVM are components. The worst is when the AQ, DT and SVM are components of the ensemble. It also appears that any ensemble with at least the AQ as a member achieves good results.

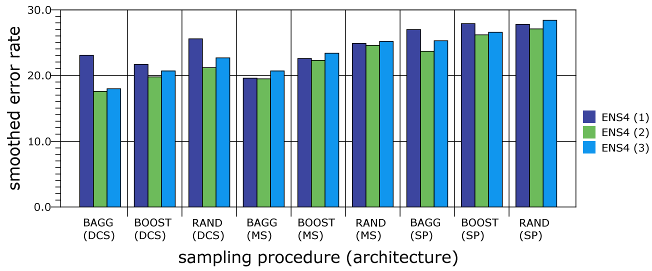

Fig. 4 Multiple classifier system (3 members)

The ensemble learning methods with two members (Fig. 5) are nearly identical to those observed for ensembles with three members. All ensembles achieve higher accuracy rates when bagging and boosting are used. Otherwise, the performance of all the methods, on average, worsens when randomisation is used. The best performing ensemble is the AQ and *k*-NN, while the worst is the combination of an ANN an AQ (randomisation). For stacking, the ensemble of an ANN, *k*-NN and NBCs proves to be the worst-performing method.

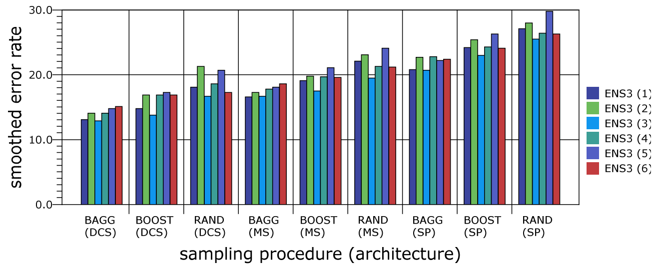

Fig. 5 Multiple classifier systems (2 members)

VI. Remarks and Conclusion

In multi-sensor data fusion, integrating multi-sensor observation data with different observation errors to achieve more accurate positioning of the target has never been so important in the information fusion community. Our paper aims to investigate the impact of multiple classifier systems for MSTT purposes for human activity identification.

Open questions related to predicting with confidence addressed include: how can data be utilised effectively to achieve more efficient confidence-based predictions using ensemble classifiers? To this end, the significant contributions of the paper include showing the robustness of single classifiers for MSTT and further show how MCL systems provide statistically significant improvements in performance over the single best classifier.

DCS that looks to segment the population in several sub-regions is a consistently good performer. All the experiments (except for the ensemble with all the classifiers as components) yield inferior to the single best classifier. However, the performance of MS combination strategies provides statistically significant improvements over the single best classifier. Ensembles with a combination of three classifiers outperformed the other MCL systems, with randomisation being a poor performer compared to other resampling procedures such as bagging or boosting. The most exciting result is bagging performance, which consistently outperformed all other multiple classifier systems, especially for ensembles with a combination of three single classifiers.

In sum, this research provides an effective and efficient approach to track targets, especially human activity. Since target tracking is a costly and lengthy process, it is often delayed because of the difficulty of tracking a target in a given environment. With the help of ensemble learning, human activity can be accurately guided early to prevent the situation from getting any worse and reduce costs associated with delayed target tracking.

Acknowledgment

The authors thank to their colleagues for their invaluable comments and suggestions when preparing the paper.

References

[1] Aha, D. W., Kibbler, D. W. & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6 (37): 37-66.

[2] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 26 (2), 123-140.

[3] Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and Regression Trees*. Wadsworth.

[4] Cervone, G., Michalski, R. S., Kaufman, K. and Panait, L. (2000). Combining Machine Learning with Evolutionary Computation: Recent Results on LEM," Proceedings of the Fifth International Workshop on Multistrategy Learning (MSL-2000), Guimarães, Portugal, pp. 41-58.

[5] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20* (3), 273–297.

[6] Cox, D. R. (1966) Some procedures associated with the logistic qualitative response curve. In *Research Papers in Statistics: Festschrift for J. Neyman* (ed. F. N. David), Wiley, New York, pp. 55-71.

[7] Dietterich, T. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomisation. *Machine Learning*, 40 (2), 139-158.

[8] Duda, R. O. and Hart, P. E. (1973). *Pattern Classification*, 2nd Edition, New York: John Wiley & Sons.

[9] Freund, Y. & Schapire, R. (1996). A decision-theoretic generalisation of on-line learning and an application to boosting. *Journal of Computing and Systems*, 55: 119-139.

[10] Finlay, S. (2011). Multiple classifier architectures and their application to credit risk assessment, *European Journal of Operational Research*, 210 (2): pp. 368-378.

[11] Ho, T. K. (1995). Random decision forests. In Proceedings of the 3rd international conference on document analysis and recognition, 278-282.

[12] Hosmer, D. W. and Lameshow, S. (1989). Applied Logistic Regression. Wiley: New York.

[13] Jolliffe, I. (1986). Principal Component Analysis. Springer Verlag.

[14] Kirk, E.E. (1982). *Experimental design* (2nd Ed.). Monterey, CA: Brooks, Cole Publishing Company.

[15] Kittler, J., Hatef, M., Duin, R. P. W. & Matas, J. (1998). On combining classifiers. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20 (3): 226-239.

[16] Kononenko, I. (1991). Semi-naïve Bayesian classifier. In *Proceedings of the European Conference on Artificial Intelligence*, 206-219.

[17] Kuncheva, L. I. (2002). Switching between Selection and Fusion in Combining Classifiers: An Experiment. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*; 32.

[18] Lewis, J. J., Nokolov, S. G., Loza, A., Fernandez, E., Canga, N., Cvejic, N., Li, J., Cardinali, A., Canagarajah, C. N., Bull, D. R., Riley, T., Hickman, D., Smith, M. I. (2006). The Eden Project Multi-Sensor Data Set. Bristol University Technical report (TR-UoB-WS-Eden-Project-Data-Set.

[19] MATLAB. (2019). *version 9.6 (R2019a)*. Natick, Massachusetts: The MathWorks Inc.

[20] Michalski, R. S. Kaufman, K. &. Wnek, J. (1991). The AQ Family of Learning Programs: A Review of Recent Developments and an Exemplary Application", Reports of Machine Learning and Inference Laboratory, George Mason University.

[21] Ouyang Z., Sun X., Chen J., Yue D., Zhang T. (2018) Multi-view stacking ensemble for power consumption anomaly detection in the context of industrial internet of things. IEEE Access 6:9623–9631

[22] Nweke, H. F., The, Y. W., Mujtaba, G. et al. Multi-sensor fusion based on multiple classifier systems for human activity identification. Hum. Cent. Comput. Inf. Sci. 9, 34 (2019).

[23] Pires I., Garcia N., Pombo N., Flórez-Revuelta F., Spinsante S. (2018) Approach for the development of a framework for the identification of activities of daily living using sensors in mobile devices. Sensors 18:640

[24] Quinlan, J. R. (1993). *C.4.5: Programs for Machine Learning*. Los Altos, California: Morgan Kauffman Publishers, Inc.

[25] Ripley, B. D. (1992). *Pattern Recognition and Neural Networks*. Cambridge University Press, New York: John Wiley.

[26] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing*, volume 1, pages 318-362. MIT Press.

[27] Safavian, S. R. and Landgrebe, D., 1991. A Survey of Decision Tree Classifier Methodology. *IEEE Trans. Syst. Man Cybernet.* **21**, pp. 660–674.

[28] Schapire, R., Freund, Y., Bartlett, P. and Lee, W., 1997. Boosting the margin: a new explanation for the effectiveness of voting methods. *Proceedings of International Conference on Machine Learning*, Morgan Kaufmann, San Francisco pp. 322–330.

[29] Sebbak F, Benhammadi F, Chibani A, Amirat Y, Mokhtari A (2014) Dempster-Shafer theory-based human activity recognition in smart home environments. Ann Telecommun 69:171–184

[30] Twala, B. (2009). Multiple Classifier Learning to Credit Risk Assessment. *Expert Systems and Applications*, 37 (2010): pp. 3326-3336.

[31] Wojtusiak, J. and Michalski, R. S. (2006). The LEM3 Implementation of Learnable Evolution Model and Its Testing on Complex Function Optimization Problems," Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2006, Seattle, WA, July 8--12, 2006.

[32] Wolpert, D. (1992). Stacked generalisation. *Neural Networks*, 5 (2): 241-259.

[33] Zdravevski E, Lameski P, Trajkovik V, Kulakov A, Chorbev I, Goleva R, Pombo, N., Garcia, N., (2017). Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering. IEEE Access 5:5262–5280.

[34] Zhang, Z.; Fu, K.; Sun, X.; Ren, W. (2019). Multiple Target Tracking Based on Multiple Hypotheses Tracking and Modified Ensemble Kalman Filter in Multi-Sensor Fusion. *Sensors*, *19*, 3118.

[35] Zhu, H., Beling, P. A., and Overstreet, G. A. (2001) A study in the combination of two consumer credit scores. Journal of the Operational Research Society, 52: 2543-2559.

[36] Zhu J., San-Segundo R., Pardo J. M. (2017) Feature extraction for robust physical activity recognition. Human-centric Comput Inf Sci 7:16.